



Déploiement de la plate-forme SlapOS dans l'environnement Grid'5000

Alain Takoudjou, Christophe Cérin, Jean-Paul Smets

► To cite this version:

Alain Takoudjou, Christophe Cérin, Jean-Paul Smets. Déploiement de la plate-forme SlapOS dans l'environnement Grid'5000. 2014. hal-00958012

HAL Id: hal-00958012

<https://hal.science/hal-00958012>

Submitted on 11 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Déploiement de la plate-forme SlapOS dans l'environnement Grid'5000

Alain Takoudjou¹, Christophe Cérin¹ et Jean-Paul Smets²

Université de Paris 13¹, LIPN, UMR CNRS 7030, 99, avenue Jean-Baptiste Clément,
93430 Villetaneuse, France

² CEO ViFiB. Site web : <http://www.vifib.com/>

{alain.takoudjou, christophe.cerin}@lipn.univ-paris13.fr, jp@nexedi.com *

Résumé Ce document explique comment utiliser la plate-forme de cloud computing SlapOS dans Grid'5000. Il détaille à la fois les points clés de SlapOS, son intégration ainsi que les points bloquants qui ont été levés afin de permettre à un utilisateur de Grid'5000 de monter une expérience avec SlapOS.

1 Présentation de SlapOS

SlapOS [SSCC11] est un système de Cloud Computing [PM09] qui fournit une plate-forme d'automatisation du déploiement d'applications. Il est basé sur le langage Python et les technologies Buildout (pour le déploiement) et l'ERP Open Source ERP5 [MdCdC08] (pour la gestion de la relation client et le catalogue des applications déployables). Ainsi, d'un point de vue conceptuel, un cloud est vu simplement comme :

- Un ERP (Enterprise Resource Planning) ;
- Un modèle de déploiement ;
- Des nœuds de traitements.

SlapOS a été initialement conçu comme infrastructure pour des applications Web, il ne s'agit pas d'un cloud spécialisé dans le calcul haute performance. Un autre point particulier de SlapOS est qu'il n'est pas basé sur la virtualisation, toutefois il est possible de déployer des machines virtuelles comme KVM par exemple mais ceci est une option. Ce choix est motivé d'une part pour éviter l'empilement de couches logicielles qui peuvent nuire in-fine aux performances et d'autre part parce qu'on peut traiter le confinement par des techniques liées au système d'exploitation.

1.1 Architecture de la plate-forme

SlapOS tourne un démon appelé SlapGrid qui est construit sur la base des idées de l'intergiciel de grille de PC BonjourGrid [CF12,ACJ09,ACJ10] qui est couplé

*. Experiments presented in this paper were carried out using the Paris 13 SlapOS Cloud experimental testbed available online at <http://slapos.cloud.univ-paris13.fr> and on the Grid'5000 testbed.

1. PRÉSENTATION DE SLAPOS

à ERP5. Cela permet de gérer la plate-forme de cloud avec facturation des services.

SlapOS est basé sur une architecture (voir la Figure 1) dans laquelle les nœuds esclaves, sont tous connectés à un nœud maître.

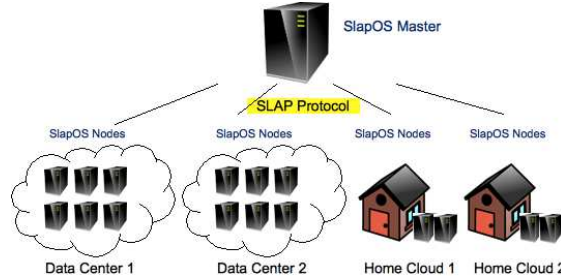


FIGURE 1. Architecture de SlapOS.

Le nœud maître appelé *SlapOS Master* ou *Master* a pour rôle l'allocation des processus c.à.d. des applications, tandis que les nœuds esclaves appelés *SlapOS Node* ou *Node* ont pour rôle l'installation et l'exécution des services. Le Master constitue un annuaire centralisé de SlapOS Nodes, c'est lui qui possède les informations qui caractérisent chaque Node. Il possède aussi le catalogue des applications qui peuvent être installées dans la plate-forme et il dispose de toutes les informations sur l'état des applications et services installés sur tous les nœuds. Ces informations sont utiles en particulier pour établir une facturation en fonction de l'usage d'un service.

Un Node, qui repose généralement sur une distribution Linux minimale, est constitué principalement d'un démon appelé SlapGrid, d'un environnement de construction et d'amorçage des applications (à base de la technologie Buildout) et d'un autre démon de contrôle des services appelé Supervisor.

Les Nodes SlapOS peuvent être dédiés (centre de données sur la Figure 1) ou bien ce sont des nœuds de bénévoles (home cloud sur la Figure 1). En comparaison avec les *clouds traditionnels*, SlapOS est basé sur une vue opportuniste qui peut être résumée comme suit. Dans son utilisation normale, les demandes sont servies par les nœuds de centres de données. Chaque fois que le nombre de demandes atteint un pic, SlapOS peut réorienter une partie d'entre elles sur les nœuds de bénévoles. Ce faisant, le système peut gagner sur deux points :

1. Il maintient un bon temps de réponse dans le traitement des demandes (notion d'élasticité) ;
2. Dans le cas de l'augmentation du nombre de clients du cloud, il est une bonne alternative pour garantir les SLA sans acheter de nouvelles machines.

Il peut ne pas sembler évident pour les bénévoles, de voir leur intérêt à participer à SlapOS. Notons que dans SlapOS, les bénévoles peuvent également être des clients du système. Dans ce cas, ils peuvent utiliser leur statut de bénévole pour l'obtention de prix avantageux.

1.2 Quelques concepts clés et caractéristiques de SlapOS

Les computer partitions La plupart des systèmes de cloud existants aujourd'hui sont basés sur la virtualisation. Elle permet une bonne isolation des applications vis à vis du système hôte, ce qui donne la possibilité d'exécuter plusieurs applications identiques ou différentes sur une même machine, sans qu'il y ait des interactions (fuite d'information). La virtualisation des ressources physiques de la machine a toujours un coût supplémentaire sur les performances de l'application. Pour éliminer ce coût, SlapOS introduit un concept appelé *computer partitions* (Figure 2), qui est un conteneur léger fournissant un environnement d'exécution des applications avec un niveau d'isolation raisonnable. Il s'agit tout simplement d'un dossier particulier associé à un utilisateur système qui a des droits particuliers ainsi que d'autres propriétés telles qu'une adresse IPv4, une adresse IPv6 et une interface réseau.

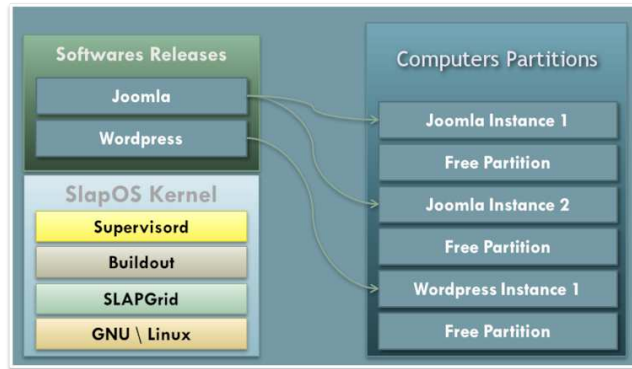


FIGURE 2. Présentation d'un nœud SlapOS.

Les Stacks et Buildout profile L'un des modes de fonctionnement de SlapOS consiste à automatiser le déploiement des applications dans la plate-forme. Pour cela, SlapOS utilise la technologie Buildout, qui est un système de compilation et d'assemblage d'applications. Le *buildout profile* est un fichier contenant un ensemble de *parts* et de *recettes* permettant de décrire pour SlapOS, et dans la syntaxe de Buildout, comment une application sera assemblée et déployée. La *part*, référencée par son nom, est tout simplement un objet ou une application manipulée par Buildout. Elle définit une recette qui décrit une logique de gestion et un ensemble de données qui seront utilisées pour la construire. La *recette* est un objet qui sait comment installer, mettre à jour ou désinstaller une part précise. L'ensemble des recettes SlapOS est disponible à l'adresse : <http://git.erp5.org/gitweb/slapos.git>. Cloudifier une application revient, en grande partie, à écrire des parts et des recettes une fois pour toute, pour une application donnée. Afin de faciliter le déploiement des applications de même type dans plusieurs instances différentes tout en promouvant la réutilisation, SlapOS définit la notion de *stack* qui est un environnement construit par assemblage de composants

2. PRÉPARATION DE L'ENVIRONNEMENT

ou d'applications permettant le déploiement d'une classe spécifique d'applications. Cette technique permet la généralisation du déploiement des applications de même type, tout en facilitant la construction de leur buildout profile. La complexité de l'intégration est donc masquée dans la stack. Comme exemple, il existe la stack *resilient* qui permet d'ajouter la résilience à une application, ou encore la stack *lamp* qui permet d'intégrer à moindre effort une application Web basée sur Apache, MySQL et PHP.

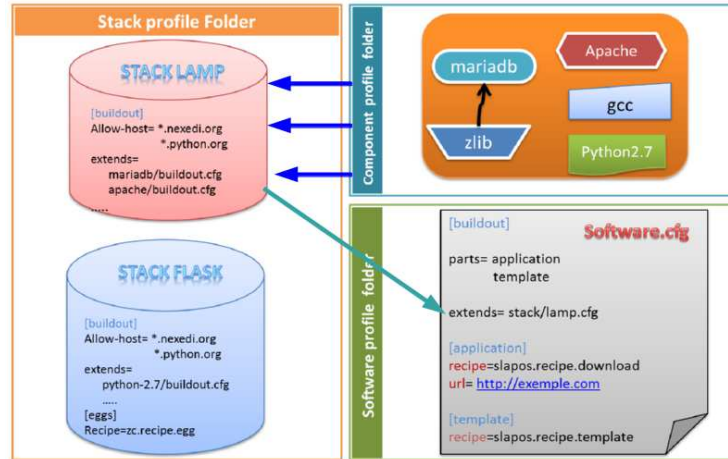


FIGURE 3. Principe d'intégration des applications dans la plate-forme SlapOS.

Les Software Releases Dans SlapOS, un *software release* (voir la Figure 3) est construit à base du profile buildout de l'application. Il contient tous les binaires nécessaires pour exécuter l'application ; ces binaires sont obtenus généralement après compilation à partir du profile Buildout. Ainsi, SlapOS est capable de déployer plusieurs instances d'une application à partir d'un seul software release dans plusieurs partitions. Ces instances sont appelées *software instances*. Chaque software instance s'exécute de manière indépendante vis-à-vis des autres. Cette technique de déploiement permet de supprimer la couche de virtualisation, ce qui permet, en théorie, un gain de performance et surtout un usage sobre des ressources d'un nœud. SlapOS peut alors tourner même sur des serveurs de faible performance. N'étant pas conçu sur les bases de virtualisation, SlapOS est quand même capable de déployer des machines virtuelles, ceci par exemple grâce à un software release résilient dédié.

2 Préparation de l'environnement

SlapOS est conçu pour fonctionner de manière native avec IPv6. L'adresse IPv6 est utilisée pour externaliser les services déployés dans les partitions. L'adresse IPv4 est locale à un Node, elle est le plus souvent utilisée lorsque deux services

2. PRÉPARATION DE L'ENVIRONNEMENT

déployés dans des partitions différentes doivent communiquer entre eux. L'IPv6 permet le déploiement aisé de SlapOS, grâce notamment à l'auto-configuration du réseau. L'avantage est de pouvoir disposer facilement d'adresses IP publiques accessibles dans le monde entier et en nombre illimité. Toutefois SlapOS peut également s'installer dans un environnement ne disposant que d'IPv4.

Puisque Grid'5000 ne dispose pas du réseau IPv6 de manière native, nous utilisons le tunnel OpenVPN pour fournir un réseau IPv6 aux nœuds que nous déployons afin de mettre en place notre architecture fonctionnelle de SlapOS. La manœuvre générique d'installation d'IPv6 est la suivante :

```
apt-get install openvpn
cd /etc/openvpn
echo "client
dev tapVPN
proto tcp
remote 176.31.103.87
port 443
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/vifib-keys/ca.crt
cert /etc/openvpn/vifib-keys/client-vifib.crt
key /etc/openvpn/vifib-keys/client-vifib.key
ns-cert-type server
verb 3" > /etc/openvpn/vifib.conf

mkdir -p /etc/openvpn/vifib-keys
cd /etc/openvpn/vifib-keys
wget http://www.slapos.org/vifib-ca.crt/getData -O ca.crt
wget http://www.slapos.org/vifib-client.crt/getData -O client-vifib.crt
wget http://www.slapos.org/vifib-client.key/getData -O client-vifib.key
/etc/init.d/openvpn restart
sleep 10 && ifconfig
```

Étant donné les limites d'accès dans Grid'5000 concernant certaines URL et la diversité des sources nécessaires pour déployer complètement notre plateforme, nous avons préparé des images contenant les paquets du SlapOS Node pré-installés. Ces images sont également configurées pour avoir IPv6 au démarrage, grâce à la manœuvre ci-dessus. Après déploiement, il suffira de vérifier qu'on a bien accès à internet en IPv6, en utilisant par exemple, la commande `ifconfig` ou `ping6 google.com`.

Pour réaliser l'installation du SlapOS Node soi-même, la procédure est la suivante :

```
sudo su
echo "deb http://download.opensuse.org/repositories/home: \
/VIFIBnexedi/Debian_7.0/ ./" | tee /etc/apt/sources.list.d/slapos.list
wget -O- "http://download.opensuse.org/repositories/home: \
/VIFIBnexedi/Debian_7.0/Release.key" | apt-key add -
```

```
apt-get update
apt-get install slapos-node
```

Dans le même ordre d'idées, les applications qui doivent être déployées sur les nœuds SlapOS sont compilées à l'avance et les binaires sont intégrés dans l'image. La raison est qu'elles proviennent généralement de plusieurs sources différentes. Il devient donc difficile de faire la compilation d'un software release dans Grid'5000. De plus, le temps nécessaire pour compiler complètement un software release, est généralement supérieure à une heure (si le cache binaire n'est pas utilisé). Nous préférons donc intégrer les applications à tester directement dans les images SlapOS pour Grid'5000, afin de réduire leurs temps de déploiement. Certains software releases pré-compilés sont également proposés à télécharger et à intégrer dans le nœud SlapOS.

3 Déploiement de SlapOS dans Grid'5000

Les images de déploiement de SlapOS pour Grid'5000 sont utilisables actuellement sur le site Grid'5000 de Lyon uniquement, car c'est le seul point autorisant le fonctionnement du tunnel IPv4-IPv6. Ces images sont basées sur la distribution Debian Wheezy en version 64 bits.

Selon les besoins, on peut effectuer deux types de déploiement :

- Déploiement all-in-one : Dans ce déploiement un Master et un Node SlapOS sont installés sur le même nœud, l'objectif étant de simplifier le déploiement. Dans ce cas, le Master SlapOS utilisé ici est le Slaproxy, c'est une version simplifiée du Master SlapOS qui est dépourvu de certaines fonctionnalités telles que la gestion des utilisateurs, la facturation, etc. Le Master Slaproxy peut être connecté uniquement à un seul nœud, ce qui signifie que cette procédure permet de créer des nœuds totalement indépendants les uns des autres du point de vue du Master SlapOS.
- Déploiement complet : Dans ce déploiement, le Master et les Nodes sont déployés sur des nœuds différents selon l'architecture de SlapOS de la Figure 1. Plusieurs Nodes peuvent être connectés au même Master car dans ce cas, c'est la version complète du Master SlapOS qui est déployée.

3.1 Déploiement avec Slaproxy

Ce déploiement se fait avec un Master fonctionnant en local. Ce Master est accessible uniquement sur le nœud sur lequel il est déployé. La raison de cette configuration est que le Slaproxy est très léger et ne gère qu'un seul Node. Dans sa version actuelle, le Slaproxy gère aussi difficilement le déploiement de plusieurs instances. À cause de ces limites, le déploiement de plusieurs instances d'une même application (ex : WordPress) ne produira pas le résultat souhaité. Il sera préférable de déployer une seule instance par nœud utilisant le Slaproxy. La procédure de déploiement de SlapOS dans ce cas est la suivante :

3. DÉPLOIEMENT DE SLAPOS DANS GRID'5000

```
# Réserver le nœud sur grid'5000
oarsub -I -t deploy -l slash_22=1+nodes=1,walltime=2:00

# Déployer ensuite l'image personnalisée de SlapOS Node local
kadeploy3 -e slapos-local -f $OAR_FILE_NODES -u atakoudjoukamdem -k

# Vérifier que les services slapos sont bien démarrées.
ssh root@HOTE.lyon.grid5000.fr #mot de passe grid5000
slapos node format -v --now
slapos node status
# Le message renvoyé est :
    slapproxy                RUNNING      pid 3010, uptime 0:06:34
    watchdog                 RUNNING      pid 3009, uptime 0:06:34
```

À ce stade le nœud SlapOS est déjà fonctionnel et nous pouvons maintenant procéder au déploiement des services. Il est configuré avec les paramètres par défaut (l'interface, le nombre de partitions, etc.), qui peuvent toutefois être modifiés si on le souhaite. Pour modifier les paramètres :

```
# Affichez l'aide sur la configuration du nœud local.
slapos help configure local

# Changer le nombre de partition (10 par défaut)
rm -r /etc/opt/slapos/*
slapos configure local --interface-name tapVPN --partition-number 30
```

Pour déployer un service déjà compilé :

```
# Demande du software release lamp
LAMP="http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/grid
-computing:/software/ajaxplorer/software.cfg"

slapos supply $LAMP local_computer

# Déploiement d'un nouveau service lamp
slapos request "My lamp PaaS" $LAMP

# Vérifier le statut et les paramètres du service
slapos proxy show --cfg /etc/opt/slapos/slapos-proxy.cfg \
--database-uri /opt/slapos/slapproxy.db --params
# Les paramètres de l'instance sont affichés :

slappart0: My lamp PaaS (type RootSoftwareInstance)
    dashboard_backend = https://[2001:470:1f14:169:c87a:1aff:fe8c:abb7]:8090
    login = admin
    password = 6ac20094
    dashboard = None
slappart1: MariaDB (type mariadb)
    url = mysqls://user:3031f48b@[2001:470:1f14:169:c87a:1aff:fe8c:52c3]:6446/db
    ip = 2001:470:1f14:169:c87a:1aff:fe8c:52c3
```


3. DÉPLOIEMENT DE SLAPOS DANS GRID'5000

```
# Pour voir les processus en cours dans les partitions:  
slapos node status
```

Il faut noter que, si l'on essaie de déployer une application dont les binaires compilés ne sont pas intégrés dans le nœud, alors le processus de déploiement ci-dessus va commencer par la compilation du software release. Nous allons expliquer plus loin, comment ajouter de nouveaux software releases compilés. Pour tester l'application, il faut ouvrir dans un navigateur l'URL [https://\[2001:470:1f14:169:c87a:1aff:fe8c:abb7\]:8090](https://[2001:470:1f14:169:c87a:1aff:fe8c:abb7]:8090) qui a été renvoyée dans les paramètres de la partition principale.

NB : Cette URL en IPv6 est utilisable n'importe où sur internet. Il convient donc de tester l'application sur un ordinateur ayant une connectivité en IPv6.

3.2 Déploiement de l'architecture complète

Ce déploiement nécessite au moins deux nœuds, l'un pour le Master SlapOS et l'autre pour le node. La Figure 4 montre l'architecture de ce déploiement.

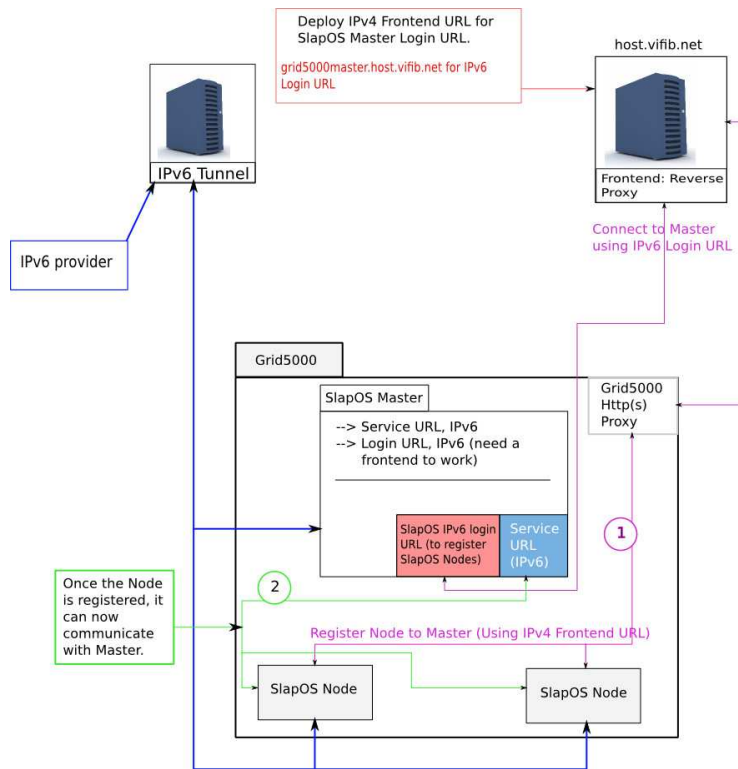


FIGURE 4. Architecture de fonctionnement de SlapOS dans Grid'5000.

Dans le schéma, nous indiquons tout d'abord que le réseau IPv6 est fourni par notre tunnel IPv4-IPv6 communiquant sur l'adresse IP 176.31.103.87 et le port

443. Seul le site de Lyon permet actuellement de déployer ce client sur les nœuds. Toutefois, les adresses IP issues du tunnel sont accessibles même en dehors de Grid'5000. Nous avons également besoin du frontend SlapOS (Proxy Reverse Server) qui nous permet d'accéder au portail Web du Master avec une adresse en IPv4, car celui-ci n'étant pas accessible via une URL en IPv6.

Lorsque le Master et les nœuds SlapOS sont déployés, ils ne communiquent pas encore car les nœuds sont inconnus du Master. Pour établir la connexion avec le Master, il faut enregistrer les nœuds. Puisque les nœuds à enregistrer appartiennent à un utilisateur, le LOGIN URL sera utilisé pour se connecter au compte de l'utilisateur, créer un *Computer* puis télécharger les certificats nécessaires pour établir la communication du node avec le Master. Cette communication initiale est indiquée par 1 dans le schéma 4. En 2 dans le même schéma, le nœud étant configuré peut communiquer avec le Master en utilisant le SERVICE URL, ce qui lui permettra par exemple de commencer le déploiement d'un service demandé par le Master.

Déploiement du Master SlapOS Afin de faciliter le déploiement du Master dans Grid'5000 une image *slapos-vifib* a été mise en place, elle configure le Master à l'aide des paramètres statiques et permet d'avoir un nœud SlapOS Master très rapidement. Pour déployer un Master :

```
# Réserver les nœuds sur grid'5000
oarsub -I -t deploy -l slash_22=1+nodes=2,walltime=20:00

# Déployer ensuite l'image personnalisée du SlapOS Master
kadeploy3 -e slapos-vifib -f $OAR_FILE_NODES -u atakoudjoukamdem -k
```

Après le démarrage de ce nœud, le déploiement du Master sera automatiquement lancé. Il faudra attendre environ 5 minutes puis vérifier que le Master est accessible, à l'aide du script ci-dessous. Dans certains cas, le Master n'est pas directement opérationnel, nous utilisons alors la commande *curl* pour vérifier qu'il est bien déployé, avant de passer à la prochaine étape. Si la commande de vérification ne renvoie pas la réponse 200, alors nous redémarrons le Master puis, nous attendons quelques minutes avant de faire une nouvelle vérification.

```
ssh root@HOTE.lyon.grid5000.fr #mot de passe grid5000
URL=https://grid5000master.host.vifib.net/
https_proxy=http://proxy:3128 curl -g -k -sL $URL \
-w %{http_code} --max-time 10 -o /dev/null
# Valeur retournée
503

slapos node restart all
sleep 200 && https_proxy=http://proxy:3128 curl -g -k -sL $URL \
-w %{http_code} --max-time 10 -o /dev/null
# Valeur retournée
200
```

3. DÉPLOIEMENT DE SLAPOS DANS GRID'5000

L'URL du service SlapOS master est `https://[2001:470:1f14:169:8850:7aff:fe6d:47dc]:10009`, elle est utilisée pour la communication entre le master et les nodes. Le portail Web du master est quant à lui accessible à l'adresse `https://grid5000master.host.vifib.net/`, cette adresse est nécessaire pour enregistrer un nœud auprès du master.

Déploiement des Nodes Après le déploiement du Master, il est nécessaire de déployer un ou plusieurs nodes. Ce déploiement est fait grâce à une image pré-configurée appelée *slapos-image*.

```
# Déploiement de l'image personnalisée du SlapOS Node
kadeploy3 -e slapos-image -f $OAR_FILE_NODES -u atakoudjoukamdem -k

# Accéder à l'image par SSH
ssh root@HOTE2.lyon.grid5000.fr #mot de passe grid5000

# Enregistrer le node
http_proxy=http://proxy:3128 https_proxy=http://proxy:3128 \
slapos node register --interface-name tapVPN \
--master-url https://[2001:470:1f14:169:8850:7aff:fe6d:47dc]:10009 \
--master-url-web https://grid5000master.host.vifib.net/ \
--login-auth --partition-number 20 "UNIQUE_NODE_NAME"
/usr/sbin/slapos-start
```

Le login et le mot de passe sont demandés, utiliser le login "demo" et le mot de passe "demo" correspondant à un compte utilisateur pré-configuré par défaut dans l'image du master. `UNIQUE_NODE_NAME` est à remplacer par le nom que nous souhaitons donner à cette machine. Pour vérifier ensuite le bon fonctionnement :

```
slapos node format --now
# Valeurs retournées
Updating computer
Posting information to 'https://[2001:470:1f14:169:8850:7aff:fe6d:47dc]:10009'
slapos successfully prepared the computer.
```

À ce niveau, nous pouvons déployer une ou plusieurs instances sur le nœud. La référence de ce node SlapOS, qui l'identifie de manière unique auprès du master se trouve dans le fichier de configuration. Cette référence nous permet de spécifier sur quel nœud nous souhaitons faire le déploiement, mais elle peut être omise.

```
ID_COMP=$(cat /etc/opt/slapos/slapos.cfg |
egrep computer_id.*= | awk '{print $3}')
LAMP="http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/grid
-computing:/software/ajaxplorer/software.cfg"
```

```
slapos supply $LAMP $ID_COMP
```

```
# Déploiement de 2 nouveaux services lamp
slapos request "My lamp PaaS 1" "$LAMP" --node "computer_guid=$ID_COMP"
slapos request "My lamp PaaS 2" "$LAMP" --node "computer_guid=$ID_COMP"
```

4. DÉPLOIEMENT DES SERVICES SLAPOS DANS GRID'5000

Au bout de 2 à 5 minutes, on peut récupérer les informations de connexion aux services déployés avec la même commande de déploiement :

```
slapos request "My lamp PaaS 1" "$LAMP" --node "computer_guid=$ID_COMP"
# Valeurs retournées
..
Instance requested.
State is : started.
Connection parameters of instance are:
{'dashboard': None,
 'dashboard_backend': 'https://[2001:470:1f14:169:3c48:45ff:feaa:dd7a]:8090',
 'login': 'admin',
 'password': '4a2669a1'}
You can rerun command to get up-to-date informations.
```

On pourra également se connecter sur le portail du Master SlapOS avec les identifiants *demo demo* pour vérifier que tous les nodes sont bien actifs et que tous les services sont déployés.

NB : Puisque les paramètres de l'image SlapOS Master sont statiques, on ne doit déployer qu'une seule image du master dans Grid'5000 à la fois. Sinon, il y aura des conflits d'adresses IP et de ports. Si on souhaite déployer un nouveau master SlapOS, il faudra déployer un service *slapos-master* sur le node SlapOS, puis, se connecter au service pour terminer la configuration. Les étapes de configuration ne sont pas automatisables et nécessitent des services externes à Grid'5000 tel que le *frontend SlapOS*¹.

Configuration manuelle du Node Dans le cas de l'architecture complète de SlapOS, l'enregistrement du Node peut aussi se faire manuellement. Pour cela, il faut créer un *computer* depuis l'interface Web du Master, copier les certificats générés dans le nouveau nœud, puis modifier le fichier de configuration de SlapOS à la main. Le document <https://slapos.cloud.univ-paris13.fr/wiki/install-slapos-packages> montre comment procéder à l'enregistrement manuel d'un SlapOS Node.

NB : Il faudra donc utiliser dans le document [https://\[2001:470:1f14:169:8850:7aff:fe6d:47dc\]:10009](https://[2001:470:1f14:169:8850:7aff:fe6d:47dc]:10009) comme étant la valeur pour `master_url`.

4 Déploiement des services SlapOS dans Grid'5000

Dans cette section, nous décrivons quelques exemples de déploiement de services SlapOS directement dans Grid'5000. Certaines applications sont intégrées dans les images SlapOS que nous avons préparées mais d'autres peuvent aussi être télécharger après le déploiement de l'image SlapOS. Nous montrons également comment sauvegarder une nouvelle image SlapOS que nous avons modifié.

1. Configuration du SlapOS Master : <http://community.slapos.org/sooe-SlapOS.Book/developer-Allocate.SlapOS.Master.Instance>

4.1 Applications intégrés dans les images SlapOS

Dans le but de permettre de tester très rapidement les services SlapOS dans Grid'5000, nous avons ajouté dans les images SlapOS Node une application permettant de déployer un environnement d'hébergement et d'administration des applications web basées sur Apache, PHP et MySQL. De ce fait, immédiatement après la mise en place de notre plate-forme SlapOS on peut tester ce service SlapOS en exécutant les commandes ci-dessous :

```
ID_COMP=$(cat /etc/opt/slapos/slapos.cfg |
egrep computer_id.*= | awk '{print $3}')
LAMP="http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/grid
-computing:/software/ajaxplorer/software.cfg"

slapos supply $LAMP $ID_COMP
slapos request "My lamp PaaS" "$LAMP" --node "computer_guid=$ID_COMP"
```

Il faudra ensuite utiliser la commande appropriée au type de d'image SlapOS que nous avons utilisé (`slapos-local` ou `slapos-image`) pour vérifier l'état de l'instance et obtenir les paramètres de connexion aux services.

4.2 Autres cas de déploiement

Depuis le site Grid'5000 de Lyon, on peut également télécharger les applications et les intégrer dans nos images en utilisant la commande *wget*. Après avoir déployé le nœud SlapOS, la manœuvre d'installation de ces applications est la suivante :

```
# Afficher la liste des software releases
ls /home/atakoudjoukamdem/public/apps/
=> boinc-client.tar.gz boinc-server.tar.gz README-vifib.txt SR_URL.txt ...

# Copie du software release BOINC Server dans le nœud SlapOS
ssh root@HOTE.lyon.grid5000.fr
cd /opt/slapgrid/
wget http://public.lyon.grid5000.fr/~atakoudjoukamdem/apps/boinc-server.tar.gz

# Extraction de l'archive
tar -xzf boinc-server.tar.gz; rm boinc-server.tar.gz

# Installer le software release BOINC Server
COMPUTER_ID="local_computer"
slapos supply SR_URL $COMPUTER_ID

# Déployer une instance BOINC Server
slapos request "My BOINC Server" SR_URL \
--node "computer_guid=$COMPUTER_ID" --parameters "force-ipv6=true"
```

Dans cet exemple, nous avons utilisé l'image `slapos-local`. De la même manière, on peut faire ce déploiement avec l'image `slapos-image`.

Dans le répertoire `/home/atakoudjoukamdem/public/apps/`, le fichier `SR_URL.txt` contient les URL (correspondant aux valeurs `SR_URL` ci-dessus) des profils correspondant à chaque applications pré-compilées et décrit également les commandes nécessaires pour déployer ces application dans Grid'5000. Lorsque les paramètres sont nécessaires pour déployer l'application, ils sont transmis via l'option `-parameters CLE=valeur [CLE=valeur]`.

D'autres applications pré-compilées pour SlapOS ainsi que les différentes images systèmes pour Grid'5000 sont accessibles à l'adresse <https://storage.host.cloud.univ-paris13.fr/>.

4.3 Sauvegarder une image SlapOS personnalisée

Si nous avons personnalisé notre image en ajouter un software release par exemple, nous pouvons souhaiter la sauvegarder pour une utilisation futur, afin de ne pas être obligé de recommencer à taper les commandes précédentes. Les commandes ci-dessous montrent comment enregistrer un environnement sous Grid'5000. Nous prenons le cas de l'image *slapos-image* :

```
# Se déconnecter du nœud si on n'est connecté pour retourner au frontend
# Sauvegarde de l'image
ssh root@HOTE.lyon.grid5000.fr tgz-g5k > my-slapos-node.tgz

# Copier le fichier de description de l'environnement puis l'éditer
cp /home/atakoudjoukamdem/slapos-image.dsc my-slapos-node.dsc
vim my-slapos-node.dsc

# Ajouter l'environnement
kaenv3 --add /home/username/my-slapos-node.dsc
```

L'édition du fichier d'environnement `my-slapos-node.dsc` a pour but d'ajuster les informations concernant la nouvelle image. Il faut changer les valeurs `name`, `version` et `author`. Pour utiliser le fichier `my-slapos-node.tgz`, il faudra également modifier la valeur `file` pour attribuer le chemin de la nouvelle image. Après l'ajout de l'environnement, l'image peut donc être déployée en utilisant le nom que nous avons donné précédemment à `name`. En supposant que notre nouvelle image s'appelle *my-slapos-node*, voici la commande de déploiement :

```
kadeploy3 -e my-slapos-node -f $OAR_FILE_NODES -k
```

Une fois que le système a démarré, on peut se connecter en utilisant l'identifiant `root` et le mot de passe par défaut qui est `grid5000`.

5 Futurs travaux dans Grid'5000

Depuis 10 ans, Grid'5000 a démontré qu'il était possible grâce au logiciel Kadeploy de mutualiser une grande infrastructure scientifique sans favoriser une technologie particulière ou une entreprise. Kadeploy est un logiciel libre développé en France et qui permet au chercheur - et non à son administration de

tutelle - de choisir librement le système d'exploitation, l'orchestrateur, le langage de programmation, le système de virtualisation ou encore la base de données avec lesquels il souhaite travailler.

Dans cet article nous avons détaillé le développement d'une image de SlapOS pour Kadeploy. Cette image peut être déployée directement sur l'infrastructure Grid'5000, sans passer par une couche de virtualisation. SlapOS nous permet ainsi d'automatiser le déploiement de services de Cloud Computing tout en bénéficiant de performances maximales grâce à l'absence de virtualisation.

Nous avons déjà réalisé une première étude de cas [WSJ14] qui explique comment déployer des serveurs de données Stork et Bitdew pour réaliser les tests BLAST (génomique). Les expériences réalisées permettent de déployer 1000 services concurrents dans des partitions SlapOS.

VIFIB a aussi développé dans le projet open source SlapOS un environnement de développement en ligne (Platform as a Service). Cet environnement peut désormais être déployé aussi bien sur un poste de travail individuel que sur l'infrastructure de VIFIB ou sur celle de Grid'5000. Il offre à l'enseignant ou au chercheur un moyen simple de définir des expériences de big data et de les déployer automatiquement sur de nombreuses machines.

Dans ce cadre, nous pouvons envisager les travaux futurs qui suivent.

5.1 Déploiement de Wendelin dans Grid'5000

Wendelin est un projet d'analyse de grands volumes de données - appelé aussi big data - né de la combinaison de *Scikit Learn* et de *NEO*. *Scikit Learn* est une bibliothèque de *machine learning* en langage python et qui remporte de très nombreux concours. *NEO* est une base de données répartie qui fournit la persistance native à des objets python. La combinaison des deux permet d'envisager le traitement en ligne de données de plusieurs téraoctets sur un cluster de machines n'ayant chacune que quelques gigaoctets de RAM. *NEO* agit alors comme une mémoire virtuelle pour *Scikit Learn* qui est donc en mesure de s'abstraire des limitations mémoire. SlapOS étant l'outil standard de déploiement de *NEO*, l'image SlapOS pour Grid'5000 permet d'envisager un déploiement automatique de *Wendelin* sur une grande infrastructure.

5.2 Pilotage de Grid'5000 via SlapOS

La demande d'un déploiement d'image sur Grid'5000 et la gestion des utilisateurs pourraient être effectués au travers de SlapOS. Ainsi, Grid'5000 pourrait fournir une interface Web simplifiée pour le déploiement d'images arbitraires et un service de comptabilisation / facturation facilitant par exemple la « re-facturation » et la comptabilité analytique (heures consommées) dans le cadre des demandes croissantes de contrôle de gestion qui s'imposent aux universités. Pour l'instant, c'est au chercheur, via la ligne de commande, de configurer l'application à la demande ou, de manière plus restrictive, de la configurer une fois pour toute dans l'image.

Dans le domaine plus industriel, piloter Grid'5000 via SlapOS permettrait de créer un « AppStore de services arbitraires » à destination des chercheurs et des entreprises, ces services pouvant être déployés indistinctement par Kadeploy ou sous forme de services SlapOS.

5.3 Déploiement totalement automatisé de SlapOS dans Grid'5000

Il s'agit ici de supprimer les quelques étapes encore manuelles du déploiement de SlapOS.

6 Conclusion

Dans ce document, nous avons montré comment procéder au déploiement de la plate-forme de cloud SlapOS dans Grid'5000. Ce déploiement est basé uniquement sur des commandes à réaliser en invite de commande ou via des scripts *shell*, ceci dans le but de permettre le déploiement répétitif, sur une multitude de nœuds de Grid'5000. Malgré l'absence d'IPv6 dans Grid'5000, comme cela est encore le cas aujourd'hui dans beaucoup d'autres institutions et entreprises, nous pouvons tout de même réaliser ce déploiement en utilisant des tunnels IPv6 dédiés pour SlapOS. Le déploiement avec Slaproxy permet de tester rapidement SlapOS que cela soit sur un ou une centaine de nœuds. Puisque la version actuelle de Slaproxy ne sait pas gérer convenablement plusieurs instances déployées sur un même nœuds, nous proposons également un déploiement qui permet de mettre en place l'architecture complète de SlapOS dans Grid'5000, en utilisant le Master SlapOS basé sur ERP5.

Remerciements : nous remercions tout particulièrement les collègues suivants de Tunis pour leur travail de commentaire et de relecture : Touraya Louati, Heithem Abbes, Walid Saad. Nous remercions également les administrateurs de la plate-forme Grid'5000 pour leur appui technique.

Références

- ACJ09. Heithem Abbes, Christophe Cérin, and Mohamed Jemni. Bonjourgrid : Orchestration of multi-instances of grid middlewares on institutional desktop grids. In *IPDPS*, pages 1–8. IEEE, 2009.
- ACJ10. Heithem Abbes, Christophe Cérin, and Mohamed Jemni. A decentralized and fault-tolerant desktop grid system for distributed applications. *Concurrency and Computation : Practice and Experience*, 22(3) :261–277, 2010.
- CF12. Christophe Cérin and Gilles Fedak. *Desktop grid Computing*. ISBN-10 : 1439862141 ISBN-13 : 978-1439862148, 2012.
- MdCdC08. Rafael Manhaes Monnerat, Rogério Atem de Carvalho, and Renato de Campos. Enterprise systems modeling : the erp5 development process. In Roger L. Wainwright and Hisham Haddad, editors, *SAC*, pages 1062–1068. ACM, 2008.

6. CONCLUSION

- PM09. Timothy Grance Peter Mell. *The NIST Definition of Cloud Computing*. NSIT : National Institute of Standards and Technology, 2009.
- SSCC11. Jean-Paul Smets-Solanes, Christophe Cérin, and Romain Courteaud. Slapos : A multi-purpose distributed cloud operating system based on an erp billing model. In Hans-Arno Jacobsen, Yang Wang, and Patrick Hung, editors, *IEEE SCC*, pages 765–766. IEEE, 2011.
- WSJ14. Christophe Cérin Walid Saad, Heithem Abbes and Mohamed Jemni. Designing and implementing a cloud-hosted saas for data movement and sharing with slapos. *International Journal of Big Data Intelligence*, 2014.